ON THE ACQUISITION OF HIGH-RESOLUTION MAPS WITH OPEN SOURCE SOFTWARE AND COMMERCIALS OFF-THE-SHELF QUAD-ROTORS

João VALENTE, Adrian GUILLEN, Oscar MANRIQUE, Mercedes ARENAL and Antonio BARRIENTOS

CAR UPM-CSIC, Technical University of Madrid, C/ José Gutiérez Abascal, 2, 28006 Madrid, Spain

Abstract. Mosaics are high-resolution images obtained aerially and employed in several scientific research areas, such for example, in the field of environmental monitoring and precision agriculture. Although many high resolution maps are obtained by commercial demand, they can also be acquired with commercial aerial vehicles which provide more experimental autonomy and availability. For what regard to mosaicing-based aerial mission planners, there are not so many - if any - free of charge software. Therefore, in this paper is presented a framework designed with open source tools and libraries as an alternative to commercial tools to carry out mosaicing tasks.

Keywords: Quad-rotors, Aerial Coverage Path planning, Mosaicing, ROS, V-REP

1 Introduction

The employment of Mini Unmanned Aerial Vehicles (MUAV) in Agriculture is growing fast. The accessibility of aerial platform for general public attracted many research areas attention such as agriculture. There are many successful studies reporting the employment of MUAV in Precision Agriculture (PA) research Zhang and Kovacs (2012).

They have been used mainly to acquire imagery from agricultural terrains. Then, those images are processed and stitched together with a dedicated software in order to made what is denoted by a mosaic. Mosaics are high-resolution images obtained from merging images from adjacent areas.

Aside from what was said, it is believed that further applications with MUAV in agricultural practices can come up.

What led many researchers to purchase their own platforms is the fact that companies exploiting imagery services with MUAV tend to be expensive. On the other hand, when MUAV are purchased they often do not comes with software

to plan the flights, neither with mosaicing software. Easily this software can achieve very high prices when acquired on the market.

Finally, working in field robotics and particularly with MUAV is a very difficult task. Although commercial quad-rotors come with many off-the-shelf features they need lots of work around and maintenance. The experiments are subject to weather conditions and other logistics issues. The time frame to do experiments is usually short. Therefore, having a tool to simulate the missions beforehand is an advantage.

In this paper a framework made up from open source software (OSS) tools is proposed. The finality of this framework is to address mosaicing-based aerial missions with market-available quad-rotors, such as those from the German company ASCTEC. The software architecture was designed based in the Robotic Operating System (ROS). Simulations have been conducted in the Virtual Robot Experimentation Platform (v-rep).

This works was divide in three modules. Since it follows a ROS-based philosophy, there are three nodes: Aerial Mission Planner (AMP), mosaicing and dispatcher. Each module addresses an individual task. Moreover, the first two are stand-alone and easier to integrate in any other aerial system. The later node is specific for v-rep and ASCTEC quad-rotors. However, it can be configured rapidly to another aerial system.

The remainder of the paper is organized as follows. Section 2 present the robotics OSS and discuss witch role they could play in PA. Section 3 introduces the optimization problem tackled in those missions. Section 4 addresses the architecture design choices and how is made up. Section 5 describes the current state of the developments and goals achieved up to now. Finally Section 6 presents the conclusions.

2 Robotic open-source software

In this section some of the OSS used in robotics research are introduced. Finally, a discussion is held about the way those tools could be used in PA and which vantages might provide.

2.1 Robot operating system

ROS is a open source framework that provides a set of tools for developing robot applications. ROS runs in Linux operating systems, although there are some initiatives to extend it to further operation systems, such as Windows and Mac Os.

It is a powerful software to control communication between different systems or robots. It can be seen as a peer-to-peer network of processes, and can work with many different programming languages, in particular C++ and Python. Indeed, the programs can be written in those languages thanks to the libraries *roscpp* and *rospy*.

ROS offers some advantages like: Free software, each user is free to make copies and share it, even modify the software; multi-lingual, allows users to program in several languages, this is possible because it use a language-neutral interface definition language (IDL) to describe messages sent between modules; thin, it is not heavy software. In each software module a file called *manifest* is defined. In this file the libraries and tools needed to develop an application are set. So ROS will be as heavy as you need.

To understand how ROS works, first we need to explain four important elements: Nodes, messages, topics and services. Nodes are processes that performs computation, each system may be composed by many of them. Messages can be of different types and with a specific structure. Messages are sent by nodes and published to a given topic. They work as connection link between nodes. The nodes can publish and subscribe to many topics. There can be a topic with more than one node publishing and subscribing on it at the same time.

Services are composed by one string name and a pair of defined type messages, one is the request and the other is the response. Nodes use services when a request-reply like communications is need. Each time a service is called, a request is sent and then the service returns a response, Quigley et al (2009).

2.2 Virtual robot experimentation platform

V-rep is a powerful simulator which includes a lot of tools, e.g., robots, sensors. Environment can be built with obstacles, persons, furniture, etc. V-rep also enable programming in several different languages. Moreover, v-rep has also a OSS licensing when used for educational purposed.

Virtual robots have their physical behavior programmed in a file called *script* and written in LUA. In v-rep all robot dynamics and control are defined in this script. All come with a standard script which can be lately edited.

V-rep enable users to manipulate virtual robots in the follow manners: standard script (or modifying the existing one), plugins, ROS nodes, among other ways. So the user can choose the one it prefers or choose according the needs it has in a particular project.

This simulator is very useful to test algorithms and approaches applied to robots before trying it in the real systems. The simulator play an important role in the development phase mainly when working with quad-rotors. Because it can avoid unexpected behaviors, robot crashes (i.e. saving lots of money) or even people injuries (i.e. avoid having legal issues).

For example, the quad-rotor that comes in v-rep by default has a well defined dynamic model, likewise an attitude and position control. Those can be edited and improved as stated before. Therefore, a new prototype can be designed in a rapid and efficient way, because the quad-rotor script and the graphical interface are directly linked. V-rep includes several ROS-Services as well, so its easy to communicate and control the virtual quad-rotor from ROS, Freese et al (2010).

2.3 Discussion: Precision agriculture meets Robotics

The conventional remote sensing (RS) tools employed in Precision agriculture (PA) practices have been overcome by novel technologies like the MUAV. Indeed, the employment of MUAV in agricultural management is becoming each day more common e.g., Valente et al (2011); Peña et al (2013). MUAV also had become a popular platform among robotics researchers due to it availability in the market, vantages when compared with other robotic platforms, as well as the challenge of working with such a novel and promising platform. Thus, both scientific research areas meet at this point.

Take into consideration the information mentioned above. Which vantages will bring some of the tools employed in robotics to the future of agricultural research? Lets focus on the tools mention in the previous subsection as case study.

For what ROS is concerning, it is clear that the main vantage is that ROS became a standard in robotics. There is a large community working on it. Which mean that there is a lots of support and new tools coming in. There are many sensor and robots drivers ready to use. Therefore, lot of tedious work with middleware can be saved, and the effective time for reaching the findings could be increased. Above all else, ROS it is completely free.

Should be said that ROS was a boom in robotics. Since ROS come up there was an increased number of communications referencing to it. The robotic research community with time is converging to the adoption of the framework. Up to now it could be said that ROS became an accepted an officially standard in robotics. ROS is attracting more researchers because it improves significantly the time and quality of research. This phenomenon could be as well beneficial for PA researchers.

There is already available many of the devices and sensors employed today in PA, e.g., GPS, IMU, cameras, laser scanner. Therefore, half of the effort is already done.

Beyond that, if industrial manufacturers think ahead, each sensor, device, or any other type of mechanism produced could have a driver developed under ROS. If this node would be provided by manufacturers, it would stimulate the market with a novel standardization. Because each driver-type node will be available and ready to use in any system, e.g. industrial tractors, hydraulic systems, other structures and mechanisms. Promoting re-usability and portability in general purpose agricultural management.

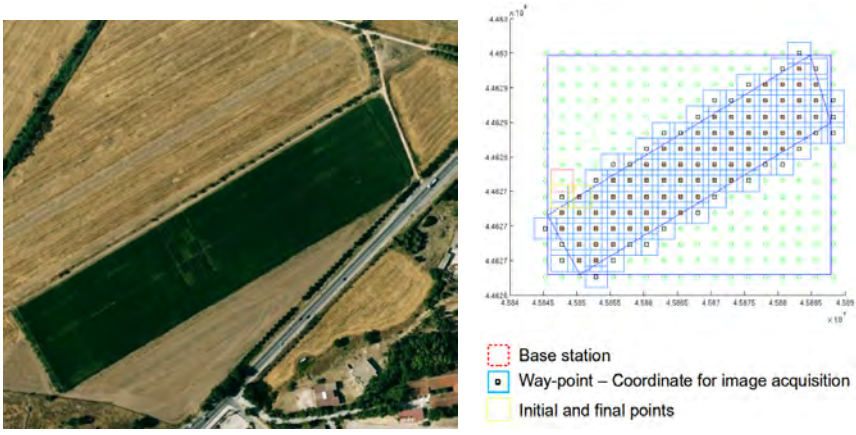
Finally, all the approaches and techniques applied in field robotic, such as surveillance, inspection, exploration, search and rescue, could be reused in agriculture, and vice versa.

3 Optimization problem

The problematic addressed herein, is the problem of surveying a determinate workspace with MUAV endowed with a digital camera. The MUAV employed

are a rotorcraft type denoted as quad-rotor. The quad-rotor fleet must cooperatively survey the overall workspace in the minimum time. In order to accomplish this task a trajectory must be computed for each quad-rotor. Such that both trajectories enable the complete coverage of the workspace.

For a better understanding, lets consider the field shown in Figure 1. The field shown is an agricultural field discretized according with the camera intrinsic parameters (focal length and sensor dimension) and imaging requisites (resolution and overlapping) applying geometric computation approaches.



(a) (b)

Fig. 1: Robots workspace: a) Real picture from the target field, b) Discretized field.

Two coverage trajectories are computed, such that the sum of both trajectories map the whole field. The mission should also be carry out in the minimum time (we can reduce the flight time if we reduce the number of revisited points by the quad-rotors, i.e., the coverage length) and in a safe way.

A safe trajectory means that the robots cannot occupy the same point simultaneously and should maintain a distance of 1 point/cell during the mission. Therefore, and summing up, there are two main security requirements (mandatory): *Collision*, the MUAV cannot be simultaneous in the same point; *safety*, the MUAV should maintain a distance from 1 point/cell of each other during the mission.

The MUAV starts and finishes the mission in a pre-defined point known as base station. In order to simplify the problematic, instead of finishing in the same point, they can finish in two different adjacent points to the base station position. See the above mentioned figure for a clear example. The green dotted line are the possible points where the MUAV can start and finish the mission.

Therefore, a complete and optimal coverage trajectory is computed iff,

Min*(number of revisited points in the environment).*

Subject to the following constraints:

1. Safe distance between robots = 1 cell
2. Pre-defined initial and final positions

4 Node-based architecture

This section proposes a node-based architecture (defined under the ROS standards). The modular architecture presented is composed by a set of nodes that can work individually, and can be easily exported to other systems. Moreover, are provided tools to carry out the complete coverage mission both in simulation, as well as in real scenarios. Figure 2 illustrate a schematic from the architecture addressed herein.

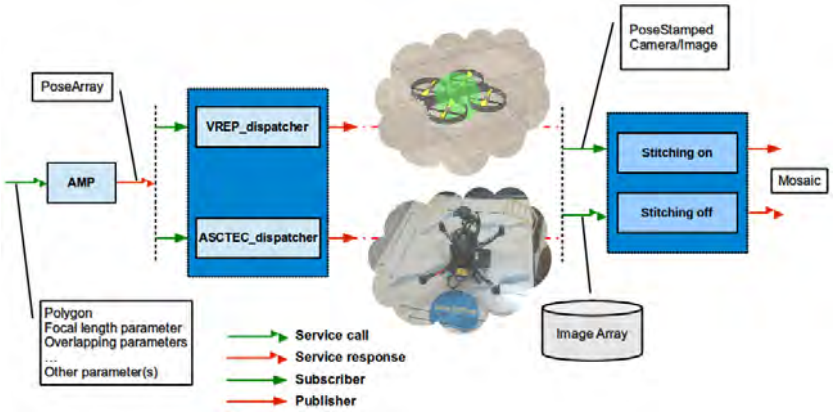


Fig. 2: Main nodes and relationships on the current architecture.

4.1 Navigation

The navigation concept might have slightly different meanings in robotics. However, in this particular case is understood as the motion of a robot over the workspace and a set of actions within it.

Thus, the idea is to have a *dispatcher* node that is able to send commands (e.g. position and orientation, orientation, shot camera) and receive (e.g. velocity, acceleration, image) data from the quad-rotor. Another, important feature of the *dispatcher* node is the versatility. It must enable switching from the v-rep simulator to the real platform each time the user desires to use one or another.

A node to control the v-rep quad-rotor was created. This node will use the ROS-Services available in v-rep, as well as user defined services. Basically, this node will be a message interpreter for and from the simulator. Usually a vector of poses (position and orientation) will be send when this service is called. Then, the robot will fly from one position to another position with a determinate orientation. In each position a photo will be taken and saved on the computer.

Like the v-rep quad-rotor, many commercial off-the-shelf quad-rotors come with an in-built position control which allows the user to send position commands, also known as way-points.

Finally, there are already some quad-rotor nodes available for ROS which handle data read from the quad-rotors and send data to it. Most of them where designed to work with the known quad-rotors from Ascending Technologies, Pelican and Hummingbird, e.g., ROS-wiki (2014).

4.2 Aerial mission planner

The Aerial Mission Planner (AMP) is another node divided in two layers. The first layer is responsible to compute the best MUAV heading, before any planning. The heading is computed in a way that the effective area of coverage is minimized. After that the field is sampled and the MUAV trajectory computed.

The second layer addresses the robot path planning. This node will have a set of input parameters, e.g. camera specifications, mission requisites, as well as the workspace borders. The output from this node is a vector of poses. In order to compute the coverage trajectory three approaches are available. One for free workspaces, and two for workspaces with forbidden zones.

With the purpose to implement the best algorithmic solution the last two algorithms have been studied and compared. An heuristic and a metaheuristic algorithm. Respectively, depth first search with a backtracking procedure, and Harmony Search (HS). Similar approaches are discussed in Valente et al (2011, 2013).

4.3 Mosaicking

Two simple ROS based mosaicking nodes were developed. An off-line node using the *OpenCV Stitcher* class and an on-line node which only use attitude given by inertial sensors and displacements in meters on the UTM coordinate system. The UAV is supposed to fly at a fair height so we can take the terrain as an approximately plain field. Errors caused by terrain elevation are ignored.

The **off-line Software** node provides a way of building a mosaic from an input formed by a number of captured images. For this purpose we used the *OpenCV* function *Stitcher*. This function automatically rectifies the images by feature detection and matching. The biggest drawback of this approach is the computational power needed in order to achieve its purpose, specially in high resolution pictures. The mosaicing off-line procedure is shown in Figure 3.

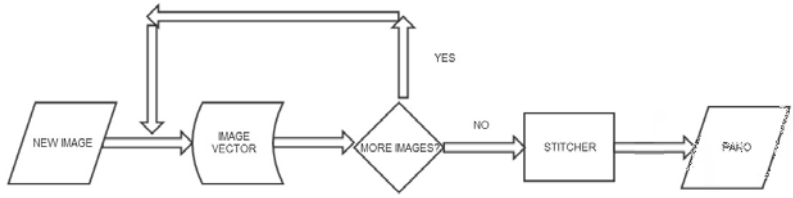


Fig. 3: Stitcher Flow

On the other hand the goal of the **on-line Software** is to use the least amount of computational power and provide a live pano output after each capture. The MUAV flies from way-point to way-point. On each way-point it takes a picture and records it positions and attitude lectures from the on-board sensors. This data is published and received by a node called *Mosaicingnode*. This node is able to process input images within a Cartesian coordinate system. This coordinate system is the most suitable to made rectifications.

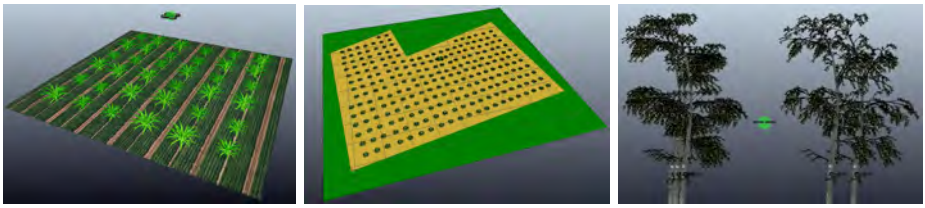
This node is subscribed to the following topics: *Counter*, *Coordinates*, and *Camera/Image*. The functions used so far are: *Mosaicing*, it is called inside the function; *Imagecallback*, it processes the new information and updates the pano; *Overlay Image*, it overlays the new image over the last pano, creating a new pano. It is called inside the function *Mosaicing*; *Imagecallback*, here the new image is received; *Coordcallback*, here the new image's parameters are received. The parameters are treated as variable type *Pose*. They consist of a position for the UTM coordinates and a quaternion for the attitude (Yaw, Pitch and Roll).

5 Ongoing work

The quad-rotor will fly over a set of determinate points (computed by the AMP) and a photo will be taken in each one (addressed by the dispatcher). The images are saved on the computer for further processing and to build the final high-resolution map (addressed by the mosaicing off-line). Or the images will be processed and saved on the computer as they are taken (addressed by the mosaicing on-line).

V-rep allows us to create the environment we want. For example we can design an environment with wide row crops, close row crops and forestry woody perennials, and even mix them. In this way with the purpose of the simulations some virtual agricultural fields have been arranged (see Figure 4).

Some simulations have been carry out with one quad-rotor in the workspace shown in Figure 4.a. Just one quad-rotor have been used for the sake of simplicity. Because, ROS must be configured for multi-robot when using more than one quad-rotor. For now there are other issues to give priority before settling the communications between several robots.



(a) (b) (c)

Fig. 4: Virtual agriculture fields: a) b) wide row crops, and c) forestry woody perennials

The quad-rotor coverage the workspace in back-and-forth motions. In this workspace forbidden zones were not set. The resulting image dataset is shown in Figure 5.

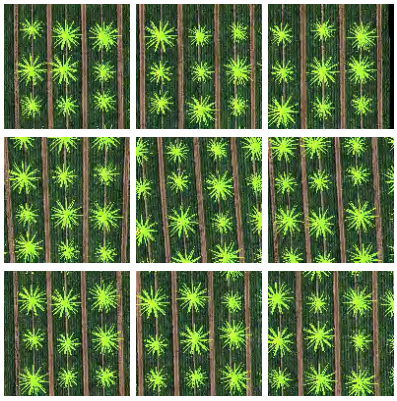


Fig. 5: Image set acquired during the coverage mission.

In this experiment the off-line mosaicing approach was tried without success. Up to now the *OpenCV* function *Stitcher* was use with minor parametrization. So it is possible that some tuning may be necessary to obtain the desired results. The next step would be configure this function so that feature detection would be focused in our areas of interest, like the overlapping zones. Should be said that this tool is mainly used for fixed cameras, which rotate and make cylindrical panographies. In spite of that, though it is possible to reduce the consumption of computational power in the stitching module pipeline, the processing time for high resolution images is still very high. Consequently, it still being studied whether this approach is feasible for this task.

That is why the best option could be the on-line method. The purpose of this approach is to decrease the computational power required in feature detection

and matching in high resolution images. However, not many tests have been done until the moment with this approach.

6 Conclusions

This paper first addresses how current robotic tools could be applied to agricultural research and the benefits it could bring to it. After that, a set of OSS tools are proposed for PA practices with commercial off-the-shelf MUAV. In particular quad-rotors. A modular software architecture is proposed with the goal to provide PA researchers a completely set of free tools for aerial mission planning, image acquisition, mosaic building, and system testing.

Acknowledgements

This work has been supported by 'Robot Fleets for Highly Effective Agriculture and Forestry Management', (RHEA) sponsored by the European Commission's Seventh Framework Programme (NMP-CP-IP 245986-2 RHEA). The authors want to thank all the project partners.

Bibliography

Freese M, Singh S, Ozaki F, Matsuhira N (2010) Virtual robot experimentation platform v-rep: A versatile 3d robot simulator. In: Proceedings of the Second International Conference on Simulation, Modeling, and Programming for Autonomous Robots, Springer-Verlag, Berlin, Heidelberg, SIMPAR'10, pp 51–62

Peña JM, Torres-Sánchez J, de Castro AI, Kelly M, López-Granados F (2013) Weed mapping in early-season maize fields using object-based analysis of unmanned aerial vehicle (uav) images. PLoS ONE 8(10):77–151

Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, Berger E, Wheeler R, Ng A (2009) ROS: an open-source Robot Operating System. In: ICRA Workshop on Open Source Software

ROS-wiki (2014) Ascending technologies pelican and hummingbird interface with ros. Http://wiki.ros.org/Robots/AscTec

Valente J, Sanz D, Cerro JD, Rossi C, Garzón M, Hernádez JD, Barrientos A (2011) Techniques for area discretization and coverage in aerial photography for precision agriculture employing mini quad-rotors. In: Proceedings of the first international workshop on Robotics and Associated High-Technologies and Equipment for Agriculture (RHEA-2011), Montpellier, France, pp 85–97

Valente J, Del Cerro J, Barrientos A, Sanz D (2013) Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. Computers and Electronics in Agriculture 99:153–159

Zhang C, Kovacs J (2012) The application of small unmanned aerial systems for precision agriculture: a review. Precision Agriculture 13(6):693–712